Network Challenges in Cyber-Physical Systems

Luís Almeida

Distributed and Real-Time Embedded Systems Lab
(DaRTES)
http://www.it.up.pt/
Instituto de Telecomunicações
Universidade do Porto
Portugal

A few application buzz words

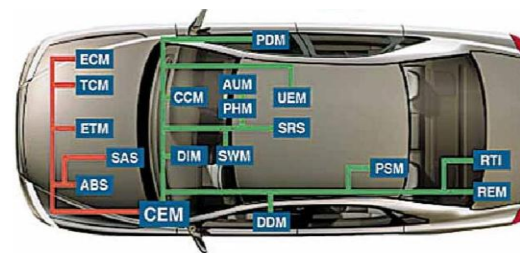
- Smart-grid, Cloud/Fog-computing, Remote interactions
- Flexible manufacturing, m2m, e-X, X-by-wire, Avionics
- Vehicular networks, Collaborative robotics
- Sensor Networks, Pervasive computing
- Ambient intelligence …

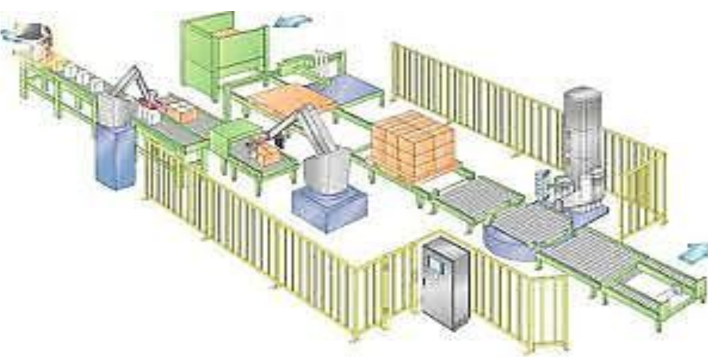
- What do these have in common?

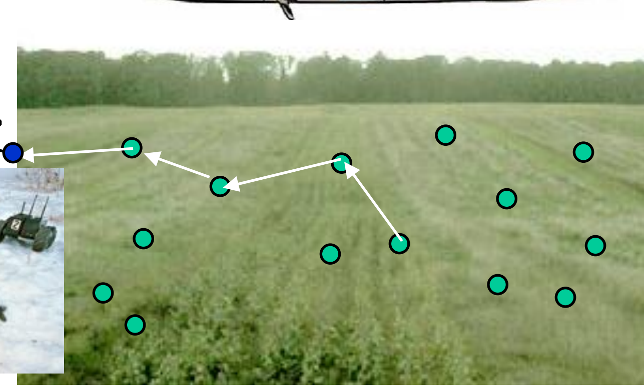
Heavily rely on networking

Many related networking frameworks

- Distributed embedded systems (DES)
- Networked embedded systems (NES)
- Ubiquitous systems (US)
- Wireless sensor networks (WSN)
- Mobile ad-hoc networks (MANET)
- ...

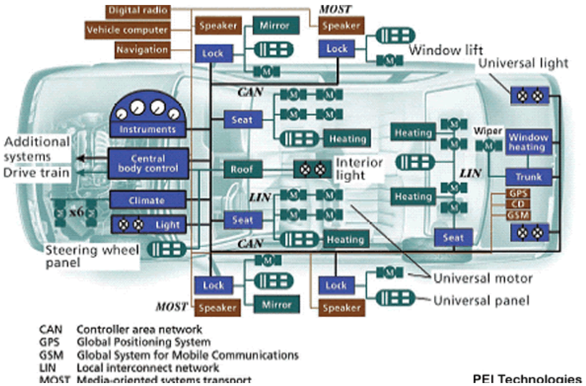






Distributed vs networked

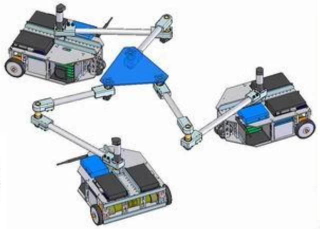
- *Distributed Embedded Systems*
  - **System-centered** (designed as a whole)
    - Confined in space (despite possibly large)
    - Normally fixed set of components
    - **Preference for wired networks**
  - Most frequent non-functional requirements
    - **Real-time**
    - Dependability
    - Composability
    - Maintainability



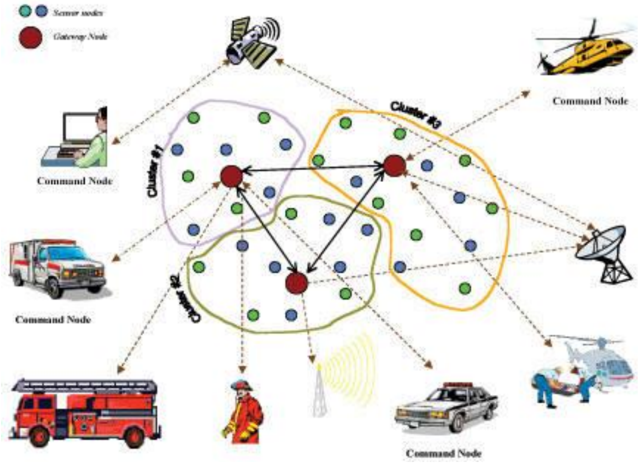
Distributed vs networked

- *Ubiquitous / Networked Embedded Systems*

  - **Communication-centered**
    (Interconnected stand-alone equipment)
    - Fuzzy notion of global system (and its frontiers)
    - **Preference for wireless networks**



  - Most common non-functional requirements
    - Scalability
    - Heterogeneity
    - Self-configuration
    - Reconfiguration
    - **(Soft) real-time**



A unified framework

Networked Embedded Systems

Networked Monitoring and Control

(Distributed) Embedded Systems

Internet of Things

- **Cyber-Physical Systems**

Ubiquitous Systems

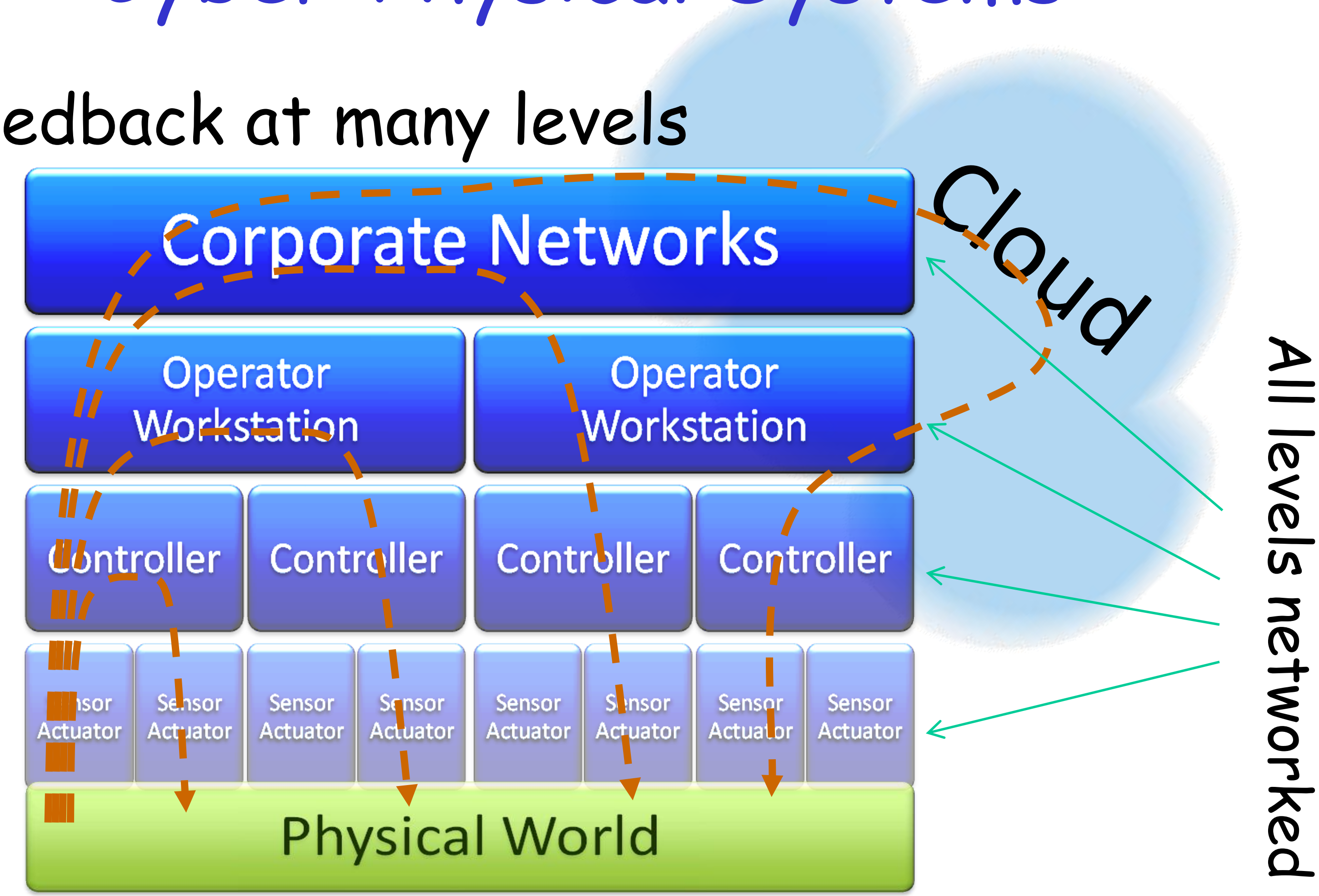
...

Towards,

better <u>physical process information</u> for better <u>control, management and planning</u>

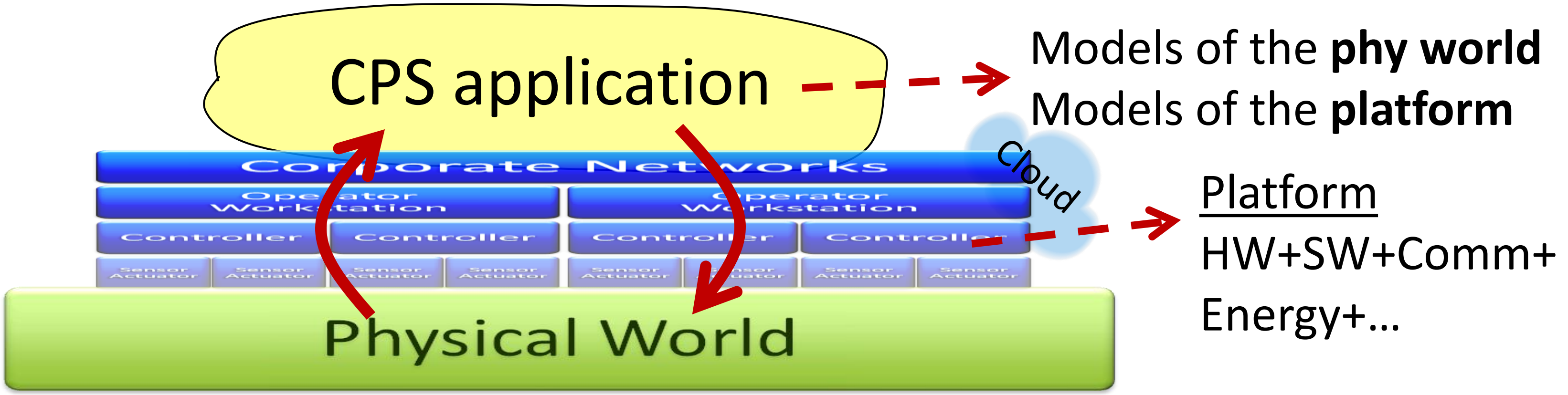
Cyber-Physical Systems

- Feedback at many levels



Cyber-Physical Systems

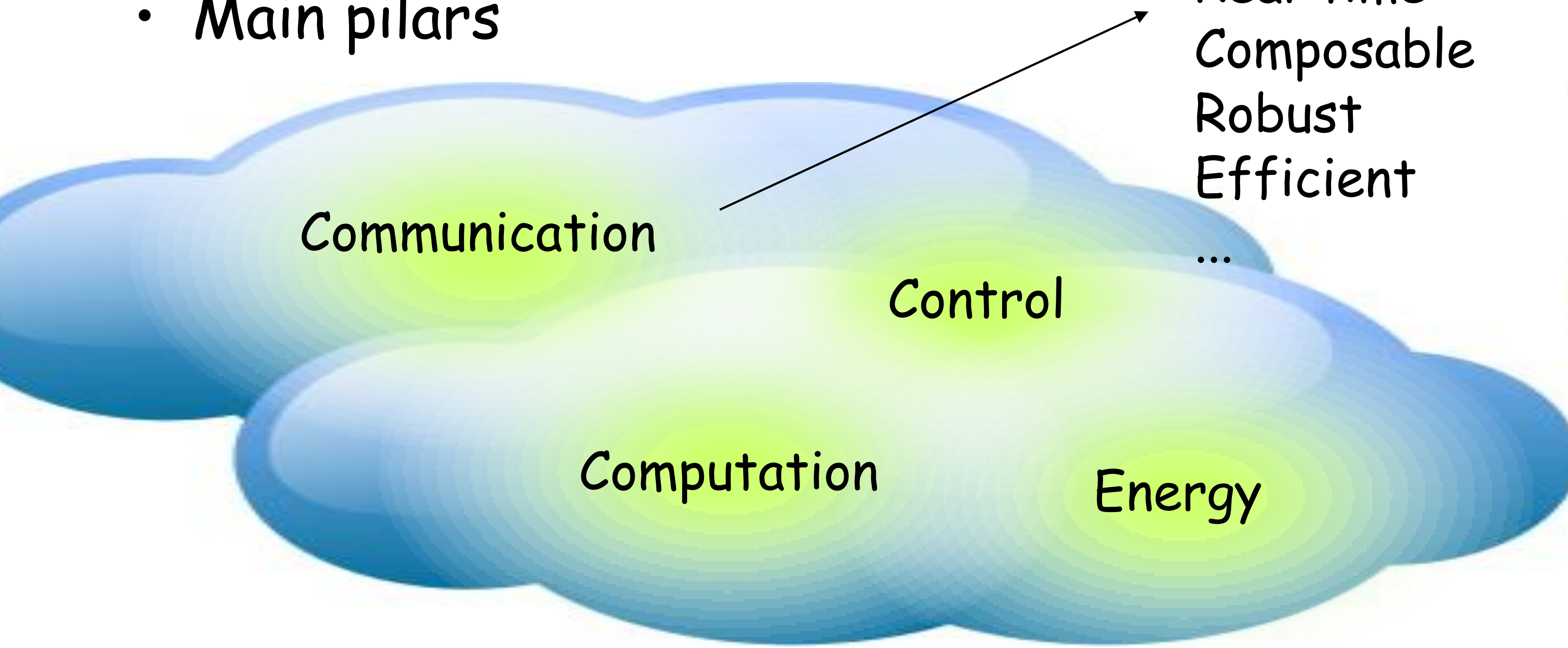
- The platform determines the degree of
  - **control** over the physical world
  - **accuracy** in the knowledge of the physical world state
- Platforms can be
  - Static, evolvable, adaptable, uncontrolable…



Cyber-Physical Systems

- Main pilars

Real-time
Composable
Robust
Efficient
...



Networks for CPS

- Are current networks adequate?
  - Real-Time communication technologies are **well developed for (static) DES**
    - Well defined system / communication requirements
  - Real-Time communication in **large networks** (Cloud/Internet) is **essentially best-effort**
    - Diversity of network service providers
      - Needed QoS-oriented protocols may not be available across providers (e.g., MPLS or RSVP-TE)
    - QoS provided mainly by differentiation of traffic classes
    - Killer applications ***(guaranteed latency matters!)***
      - Smart grid, remote interactions, collaborative robotics, ...

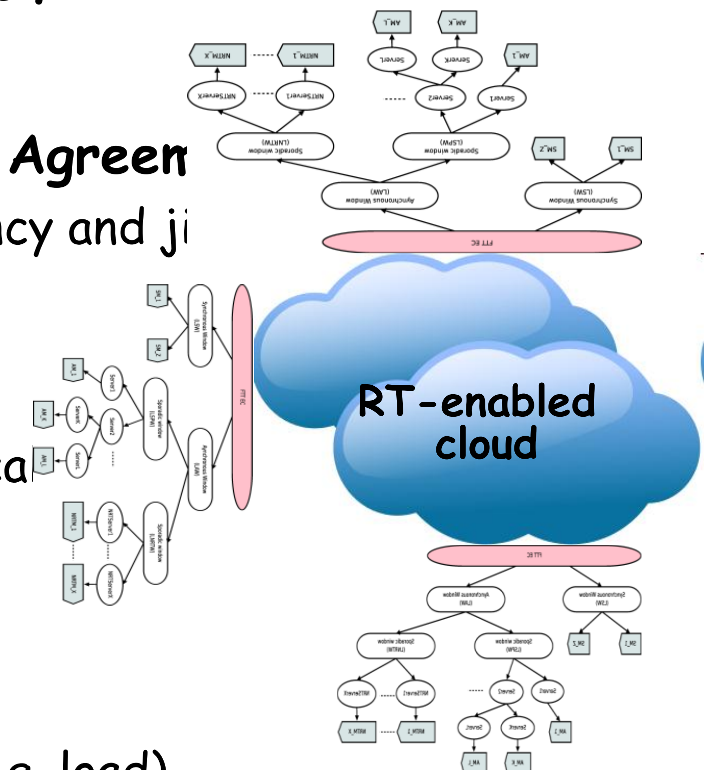
A Network Challenge for CPS

- Can we provide a **channel abstraction** that
  - is **easier to work with** for higher layers
    - Virtual channels with clear and simple interface
  - offers a **minimum guaranteed QoS**
    - Virtual circuit switching (IntServ style)
  - is **scalable** to large networks (Internet scale?)
  - considers **adaptation** to use bandwidth efficiently
    - Exploiting timing constraints and playing with slack
  - is **resilient to overloads** /cascaded failure / DoS
    - Avoiding thrashing

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

A Network Challenge for CPS

- We need a principled way of
  - Establishing
    - **Time-oriented Service Level Agreem**
      - Bandwidth + end-to-end latency and ji
  - Enforcing
    - **Resource Reservations**
      - Virtualization with hierarchica
      - Composable interfaces
      - Dynamic and adaptive
  - Using
    - **Local resource information** (e.g. load)
    - **Time-oriented traffic scheduling**
      - e.g., deadline-based



A Network Challenge for CPS

- Two views
  - **Internet networking community** has focused mainly on **scalability and throughput**, not so much on latency → solutions based on OSI layer 3+
  - **Embedded networking community** has focused on **latency** without much consideration for scalability and throughput → solutions based on OSI layer 2

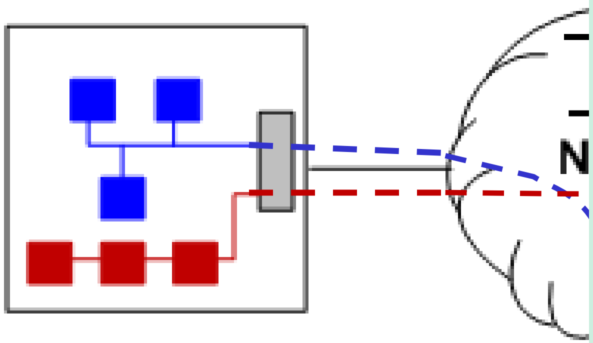
- We need a **unifying effort** (from L2 up) towards

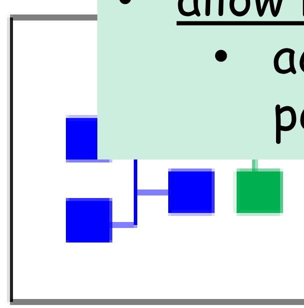
Scalable, open, efficient real-time communication

and focusing on **access networks**

Our recent related work

- **Building networks** … channel abstraction…





Dynamic virtual channel reservations

- simple composable channel interface
  - Capacity(B), deadline, period, jitter
    - **BW** = capacity/period
    - **latency** ≤ deadline
    - **latency(WC-BC)** ≤ jitter
- hierarchical channel composition
  - channels of channels
- allow ranges in interface declaration
  - acceptable vs desireable performance levels

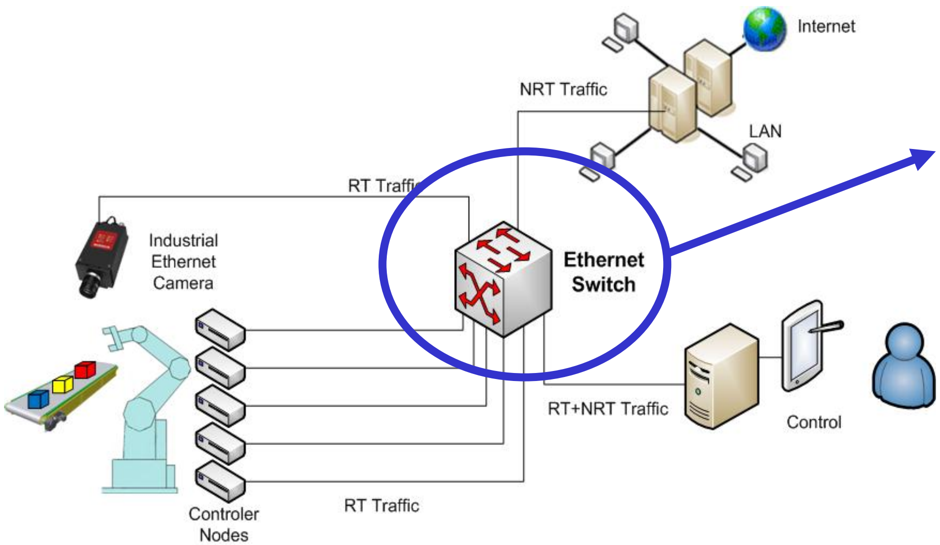
On Ethernet networks

- **The network element**
  - Any COTS Ethernet switch
- Approach:
  - **Control load submitted** to the switch
    - Allows overriding switch limitations
  - The **Flexible Time-Triggered** paradigm
    - Isochronous / asynchronous traffic
    - **Any on-line traffic scheduling** supported

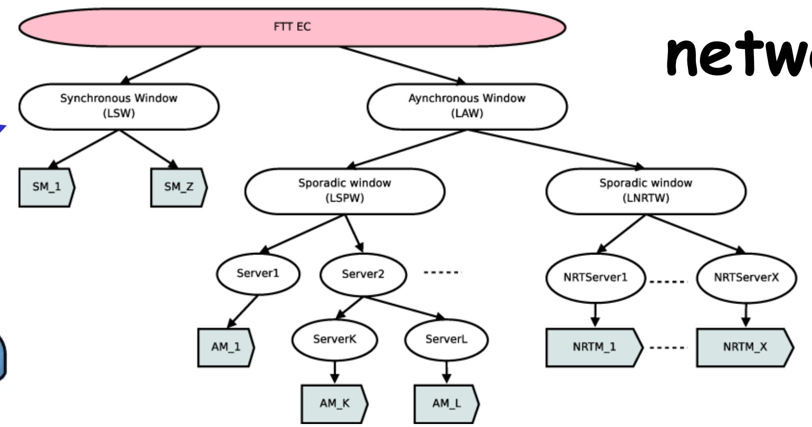
http://www.fe.up.pt/ftt

Flexible Time-Triggered Sw Ethernet

- Apps with tight/heterogeneous requirements
  - Automation / Transportation Systems / …



A virtualized network

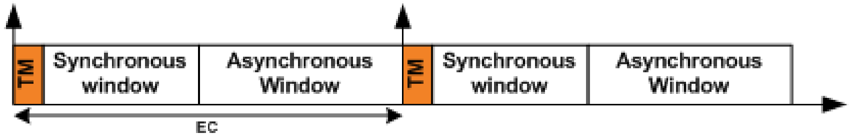


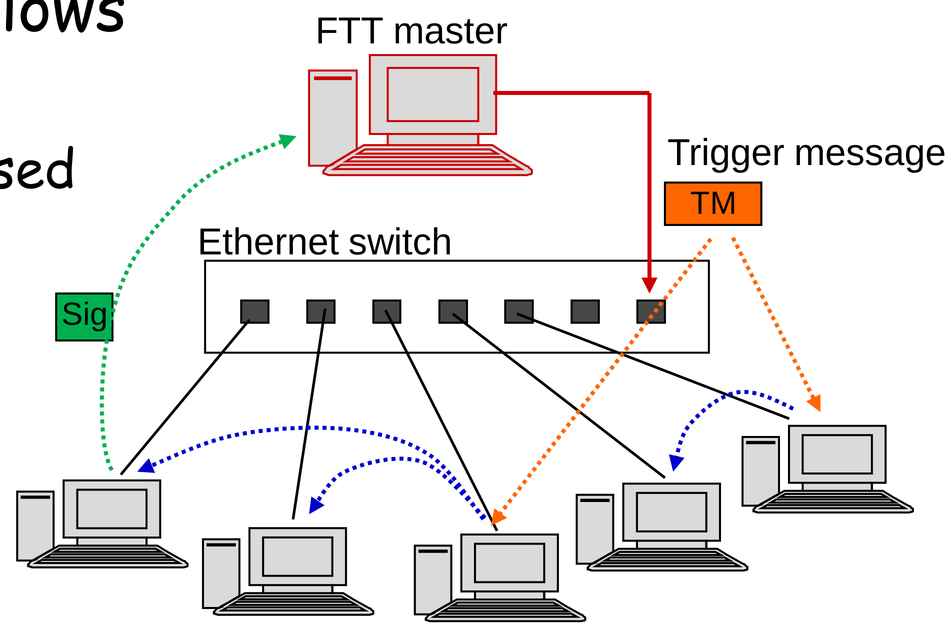
Virtual channels

- **Virtualization** with hierarchical servers (channels)
- Real-time **reconfiguration/adaptation**
- **Safe connection of components** (avoiding DoS)

FTT-SE internals

- ✓ Schedules traffic per **cycles**
- ✓ Submitting, each cycle, the **traffic that fits,** only
- ✓ **Eliminates** memory overflows
- ✓ Supports any scheduling
  - ✓ Full priorities, deadline-based Server-based, …



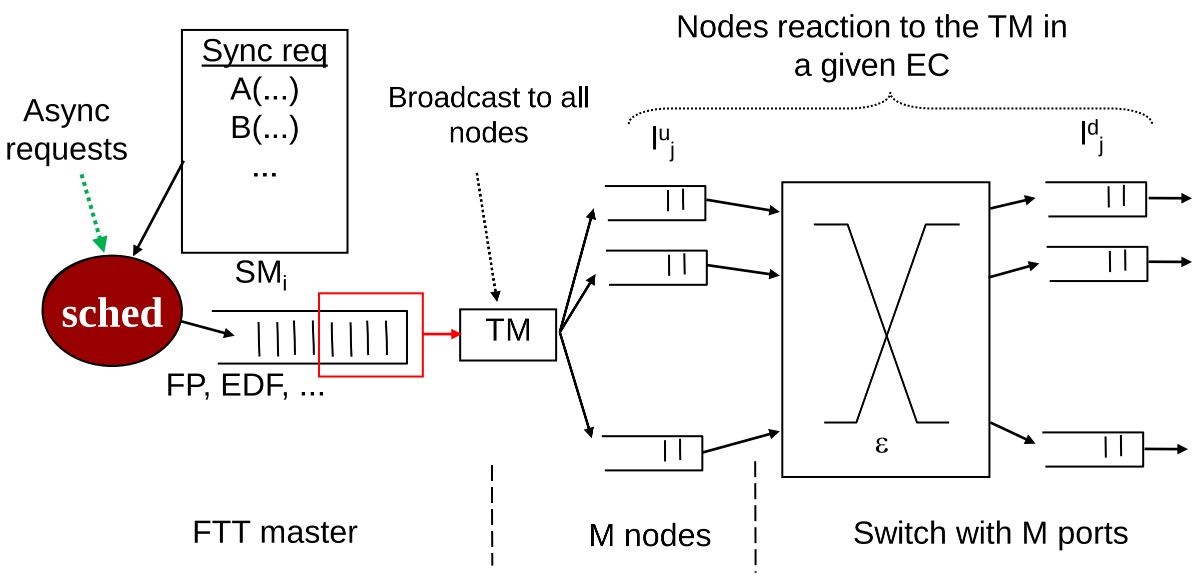


FTT-SE traffic scheduling

✓ Integrated scheduler for all traffic types

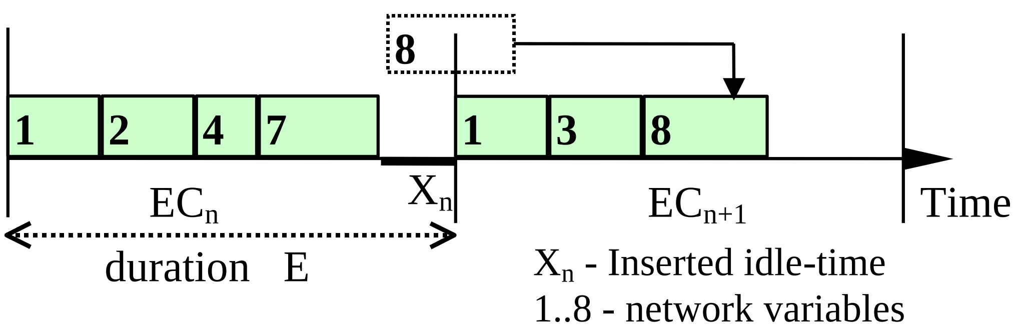
Sync: $SRT = \{SM_i: \mathbf{SM_i(C_i, D_i, T_i, O_i, Pr_i, S_i, \{R^1_i .. R^{k_i}_i\})}, i=1..N_S\}$

Async: $ART = \{AM_i: \mathbf{AM_i(C_i, D_i, mit_i, Pr_i, S_i, \{R^1_i .. R^{k_i}_i\})}, i=1..N_A\}$



FTT-SE traffic scheduling

- Basic scheduling model:
  - **Schedule within partitions** with **strict time bounds**
  - Use **inserted idle-time** (X)
    - There is **no blocking**
    - ***Any analysis for preemptive scheduling*** can be used with **inflated transmission times** (C')



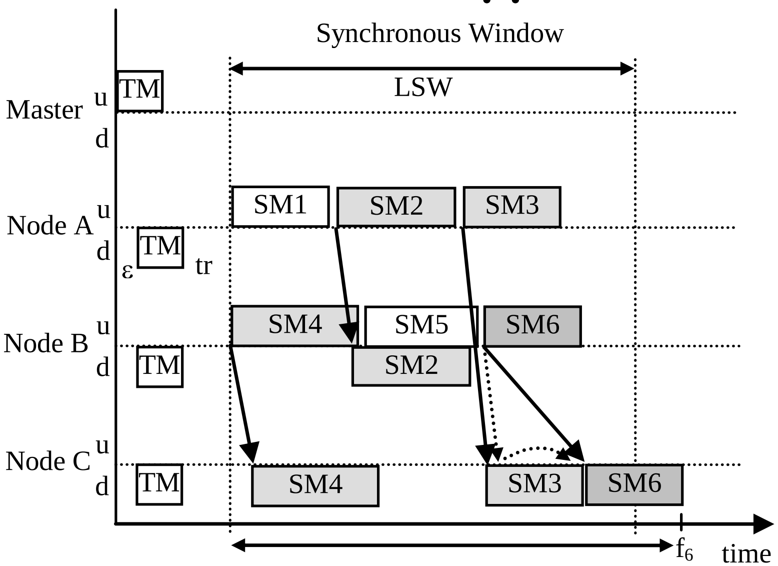
$$C_i^{'} = C_i * \frac{E}{E - X_{max}}$$

Inserted **idle-time** compensation factor

We will consider **C'** as **C** and use **preemptive analysis** in the following

FTT-SE traffic scheduling

- Interference in the uplinks appears at the downlinks as **release jitter** (J)
- **Utilization bounds** for **on-line QoS management**
  - To be **applied to each link separately**

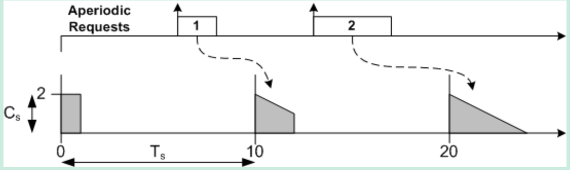
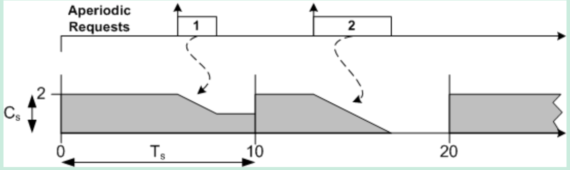
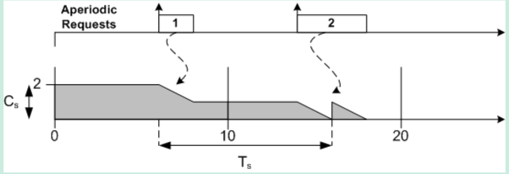


$$\forall_{i=1..n} \sum_{j=1}^{i} \frac{C_j}{T_j} + \frac{\max_{j=1..i} J_j}{T_i} \le U^{lub}_{RM,EDF}(i)$$

$$\sum_{i=1}^{n} \frac{C_i}{T_i} + \frac{\max_{i=1..n} J_i}{T_1} \le U^{lub}_{RM,EDF}(n)$$

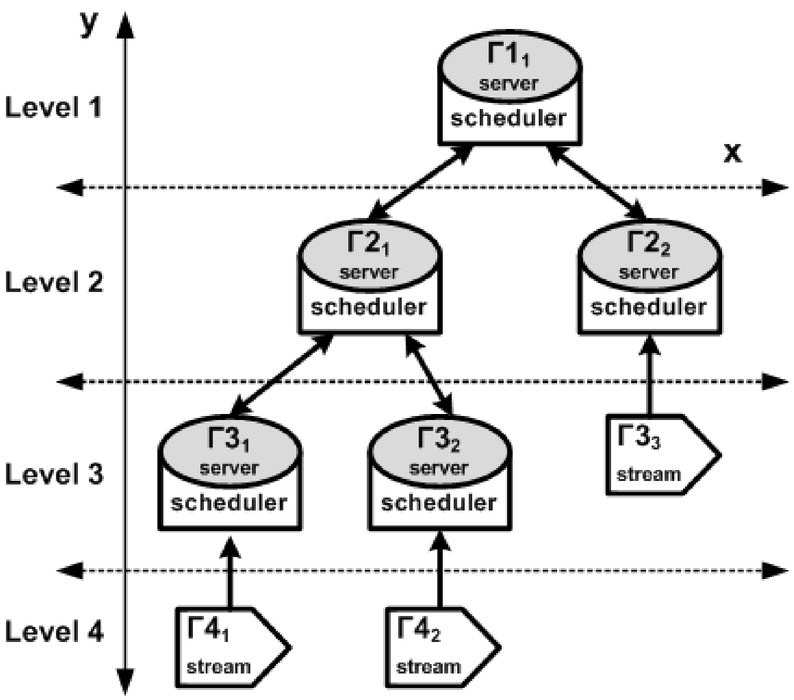
Hierarchical server-based sched.

- Support **component-based** design techniques:
  - Provides efficient **resource distribution**
  - Provides **composability** based on temporal isolation
  - Easily supports **dynamic** environments

Polling server	Deferrable Server	Sporadic Server
		

Hierarchical server-based sched.

- **Servers:** $Srv_{y_x} = (C_{y_x}, T_{y_x}, Mmax_{y_x}, Mmin_{y_x}, P_{y_x}, RT_{y_x})$
- **Streams:** $AS_{y_x} = (C_{y_x}, Tmit_{y_x}, Mmax_{y_x}, Mmin_{y_x}, P_{y_x}, RT_{y_x})$



C_{y_x} - Max message Tx time

T_{y_x} - Period

$Mmax_{y_x}$ - Tx time - largest packet

$Mmin_{y_x}$ - Tx time - smallest packet

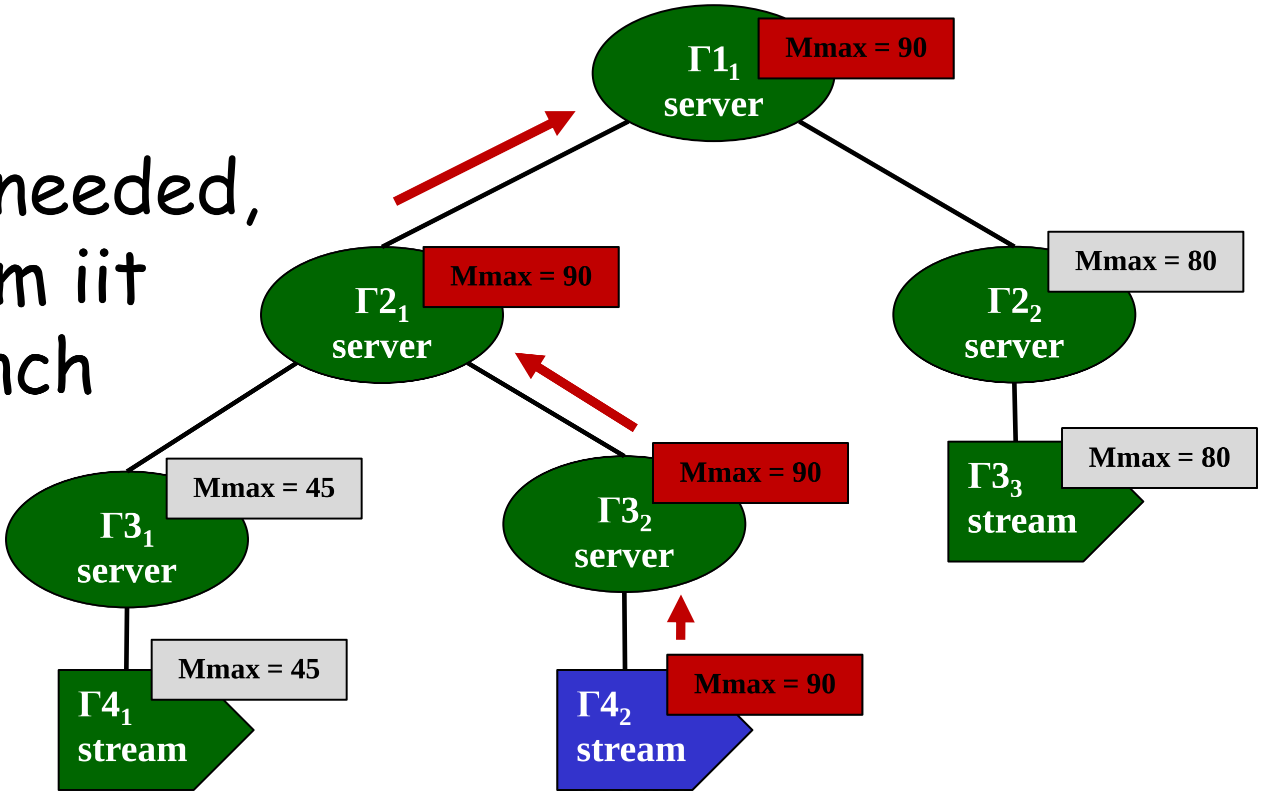
P_{y_x} - Parent component (server)

RT_{y_x} - Response time of each component

Response time analysis

First phase

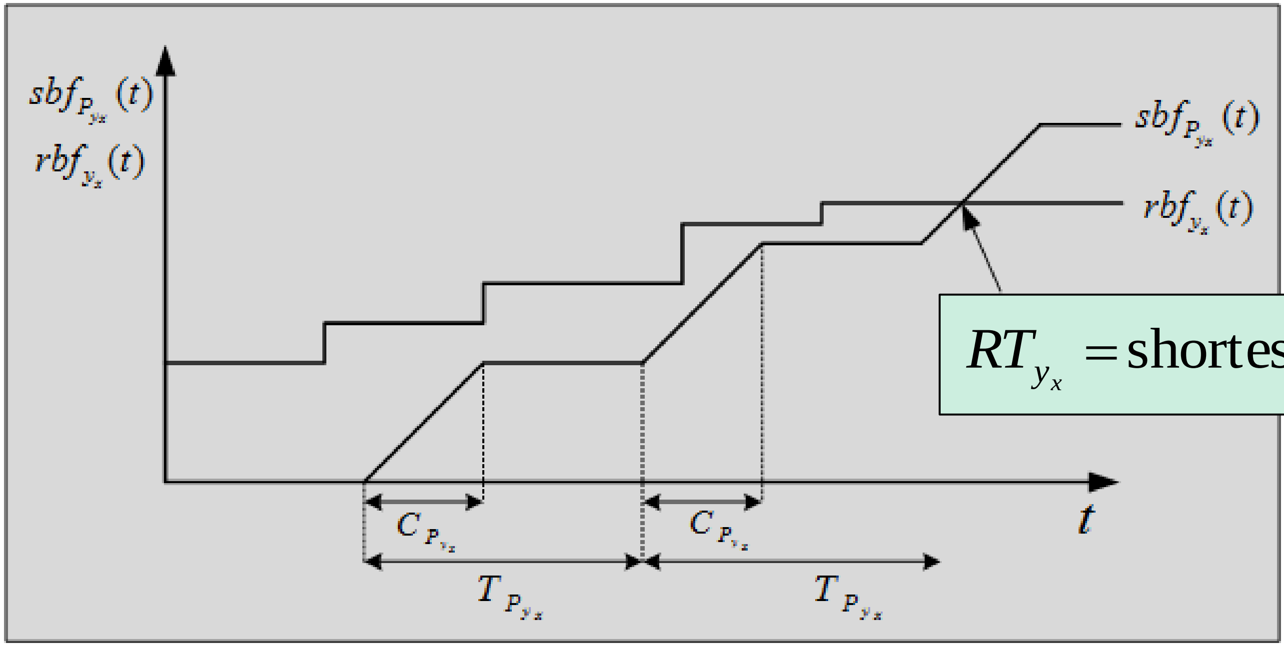
- updates, if needed, the maximum iit in each branch



Response time analysis

Second phase

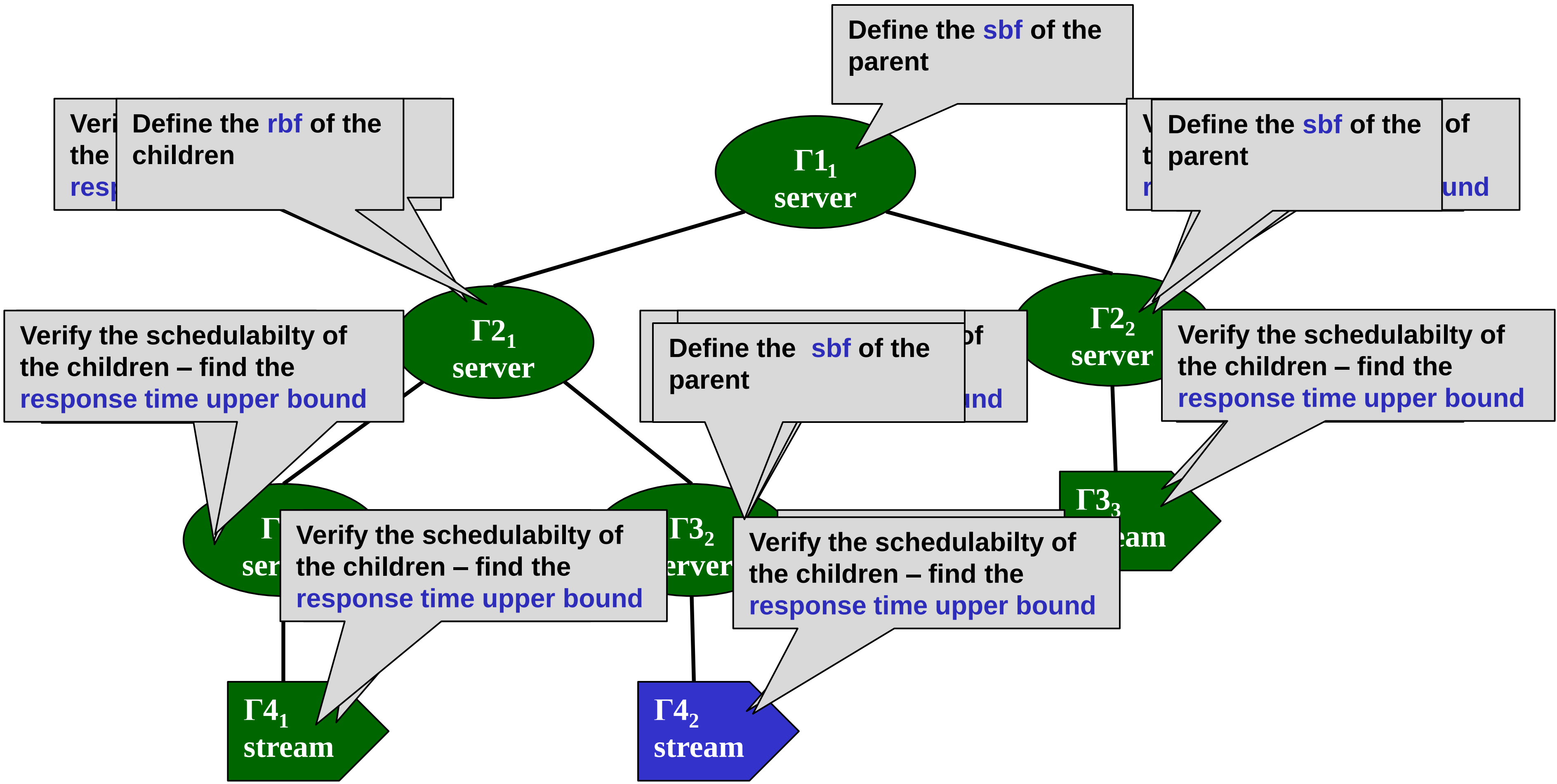
- top to bottom, check: $RT_{y_x} \leq T_{y_x} \Rightarrow \Gamma_{y_x}$ is schedulable



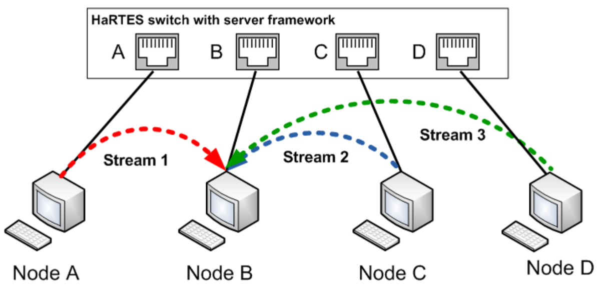
- min service
- max demand

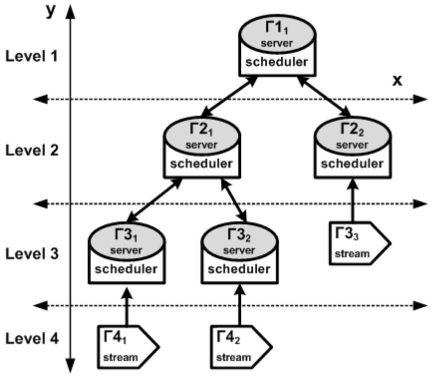
$$RT_{y_x} = \text{shortest } t^* : rbf_{y_x}(t^*) = sbf_{P_{y_x}}(t^*)$$

Response time analysis

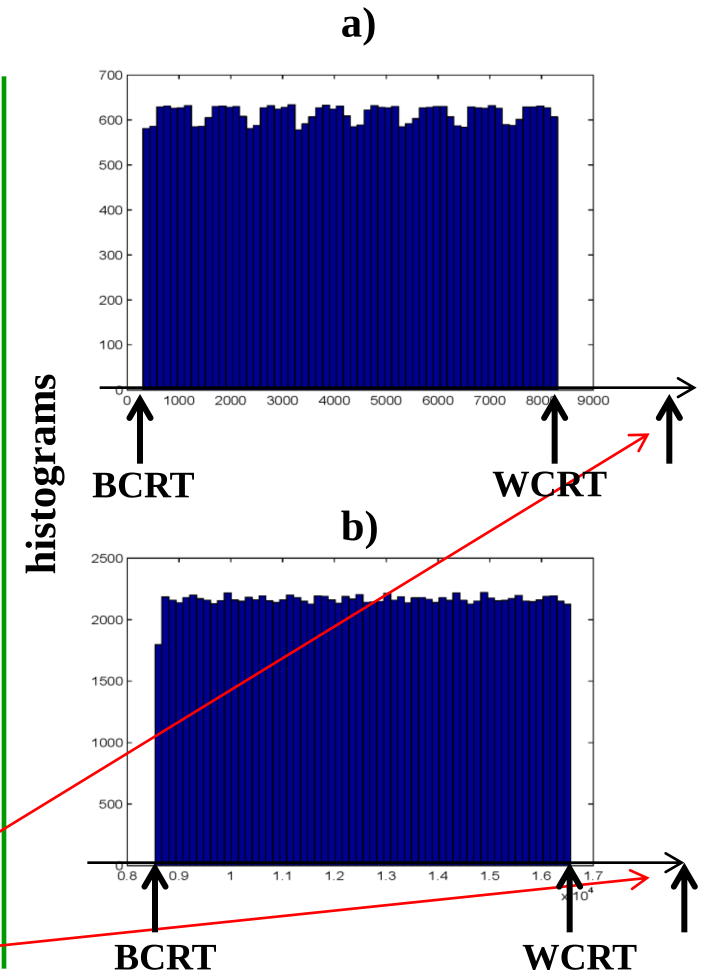


Example of response time analysis

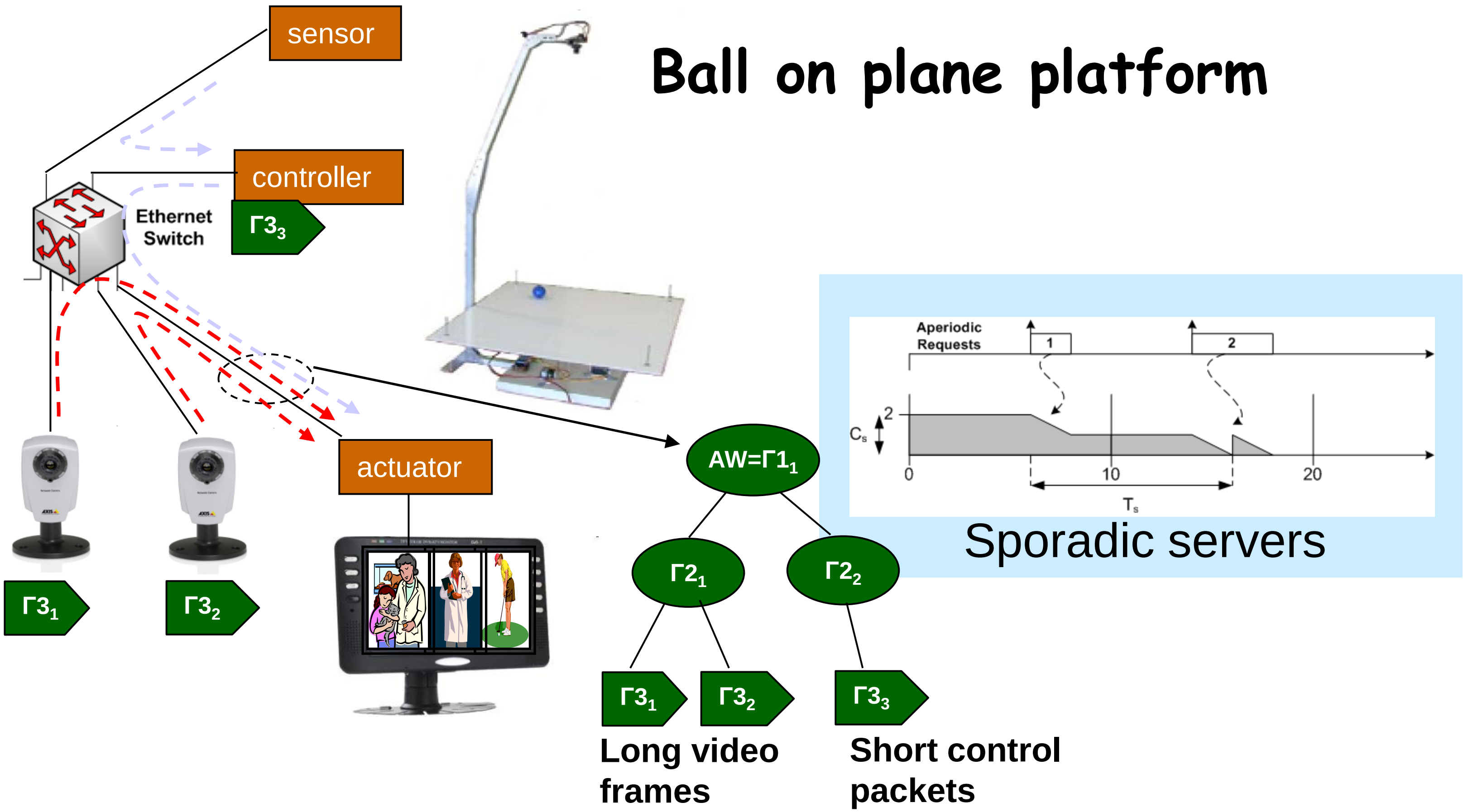




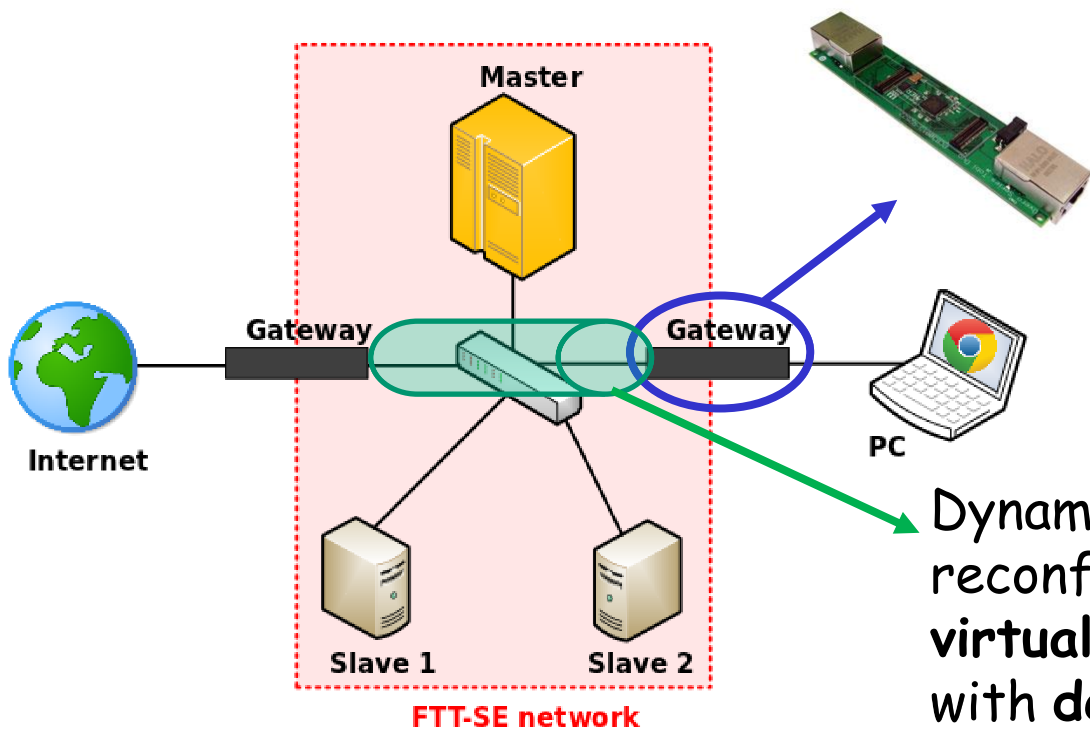
	C	T - Tmit	P	Mmax	Mmin	RT
Γ_{1_1}	700	1000	-	50	25	-
Γ_{2_1}	400	3000	Γ_{1_1}	50	25	1000
Γ_{2_2}	250	2000	Γ_{1_1}	25	25	650
Γ_{3_1}	100	8000	Γ_{2_1}	50	50	3650
Γ_{3_2}	250	4000	Γ_{2_1}	25	25	3550
Γ_{3_3}	25	2200	Γ_{2_2}	25	25	2200
Γ_{4_2}	25	7100	Γ_{3_2}	25	25	7100
$\Gamma_{4_1} a)$	50	25000	Γ_{3_1}	50	50	11550
$\Gamma_{4_1} b)$	150	25000	Γ_{3_1}	50	50	19600



Example of temporal isolation

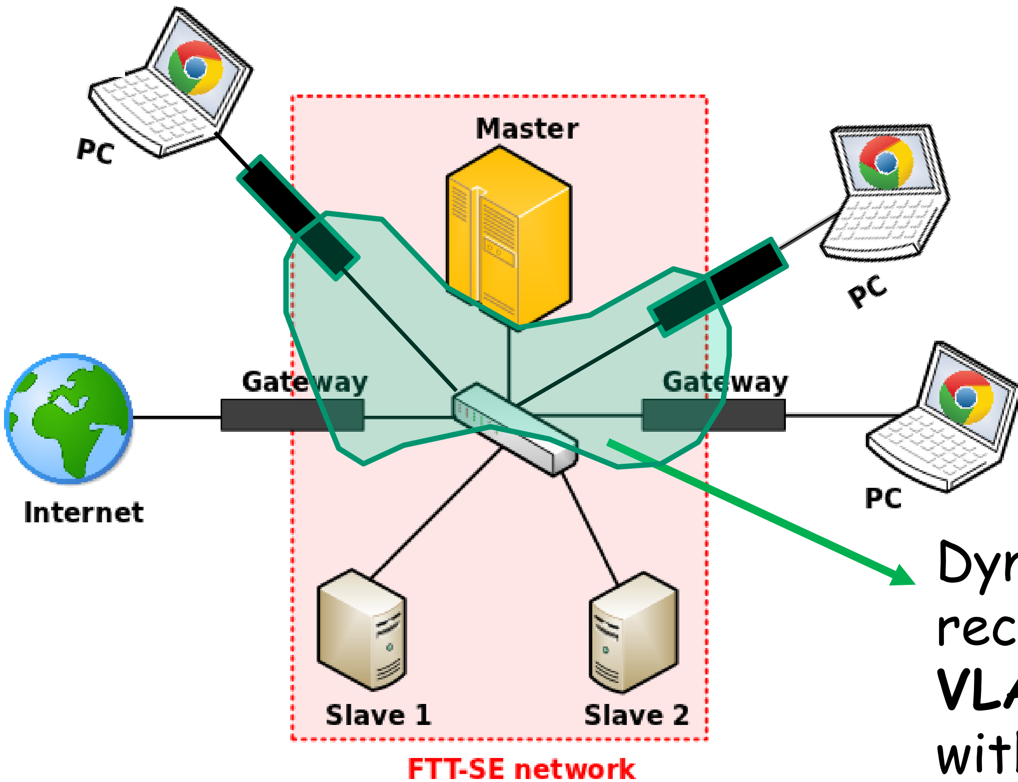


Example of temporal isolation



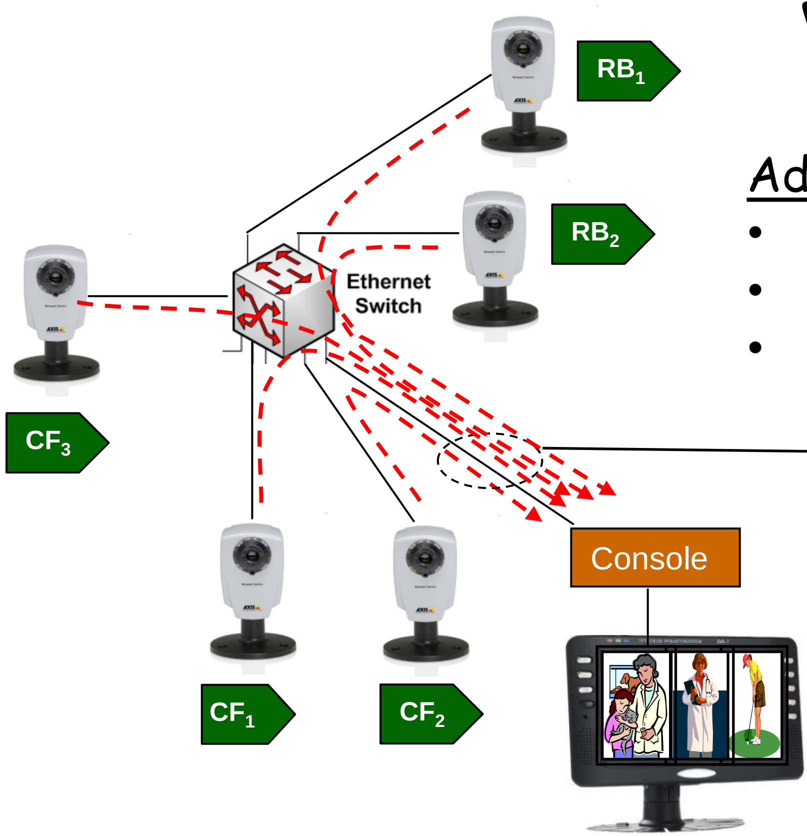
Dynamically reconfigurable **virtual channel** with **definable temporal properties**

Example of temporal isolation



Dynamically reconfigurable **VLAN** with **definable temporal properties**

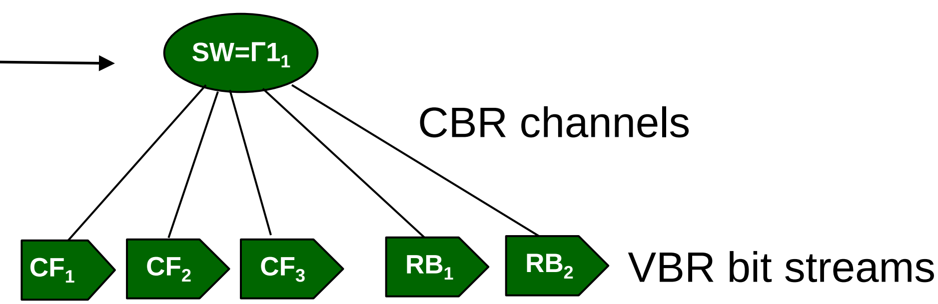
Example of adaptation



Video surveillance system

Adapting the CBR **channels bandwidth**

- exploit **BW released** by streams that are off
- reduce the use of **too strong compression**
- upon operator **request**



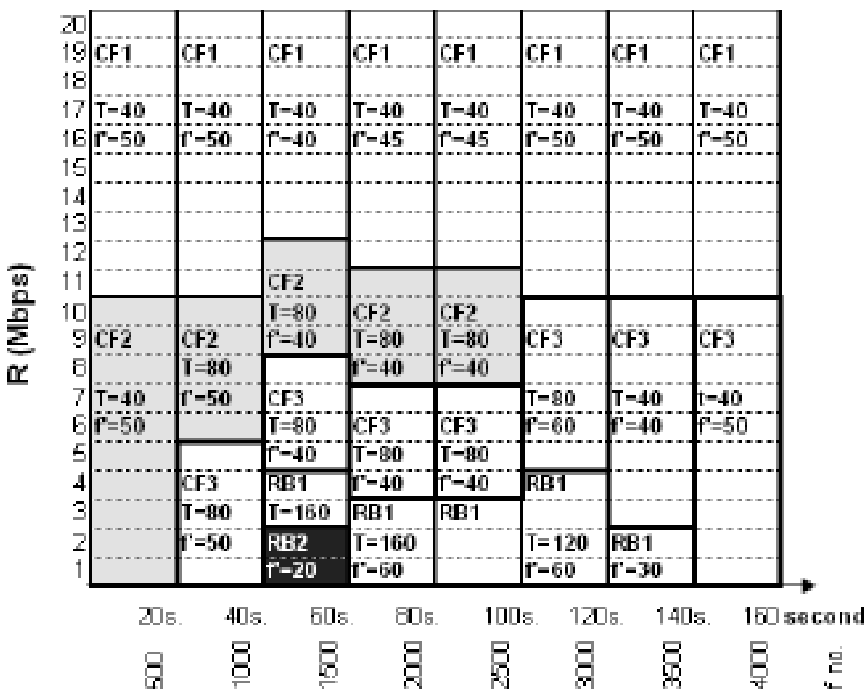
5 MJPEG cameras with variable QoS

Frame period (ms) {40, 80, 120, 160}
Frame size (kB) [30..60]

Adaptive video surveillance system

Adapting multiple CBR channels

- Streams are not always ON
- Maximize total BW usage

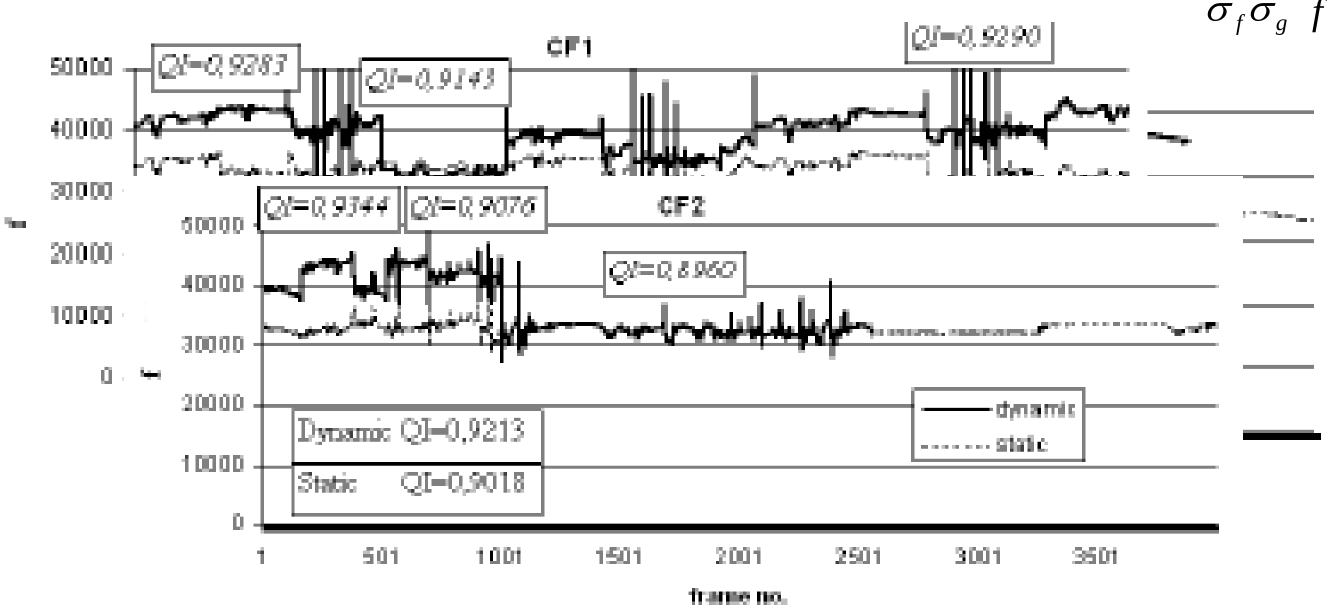


Adaptive video surveillance system

Adapting multiple CBR channels

- Evolution of the Quality Index (QI) comparing to statically allocated channels

$$QI = \frac{\sigma_{fg}}{\sigma_f \sigma_g} \frac{2\hat{f}\hat{g}}{\hat{f}^2 + \hat{g}^2} \frac{2\sigma_f \sigma_g}{\sigma_f^2 \sigma_g^2}$$



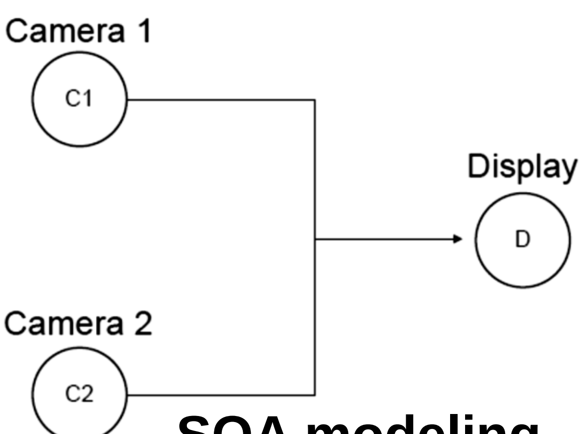
Adaptive video surveillance system

5 Mbit/s		1 Mbit/s		
V	after 20s	V		6Mbit/s at all times
1 Mbit/s		5 Mbit/s		

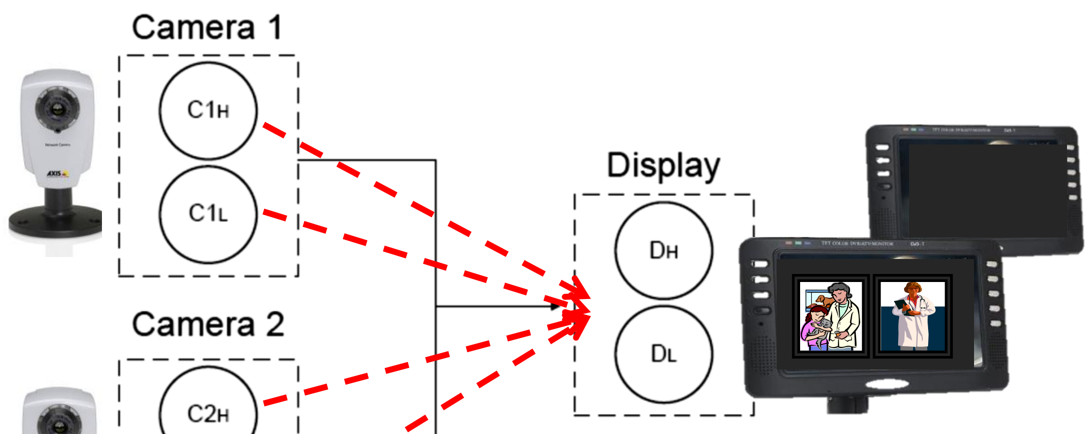
Examples of reconfiguration

http://www.iland-artemis.org/

iLAND: Service-oriented real-time middleware for deterministic and dynamically reconfigurable applications



SOA modeling

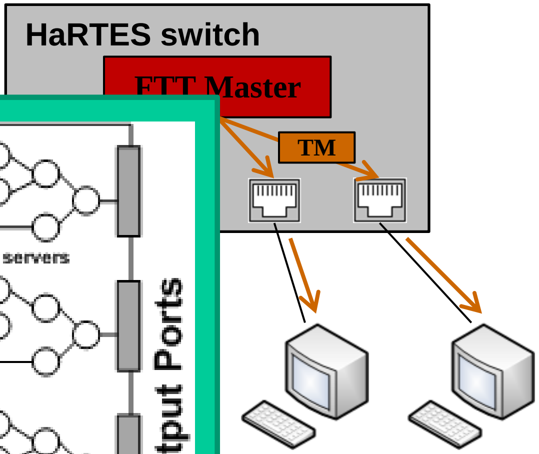


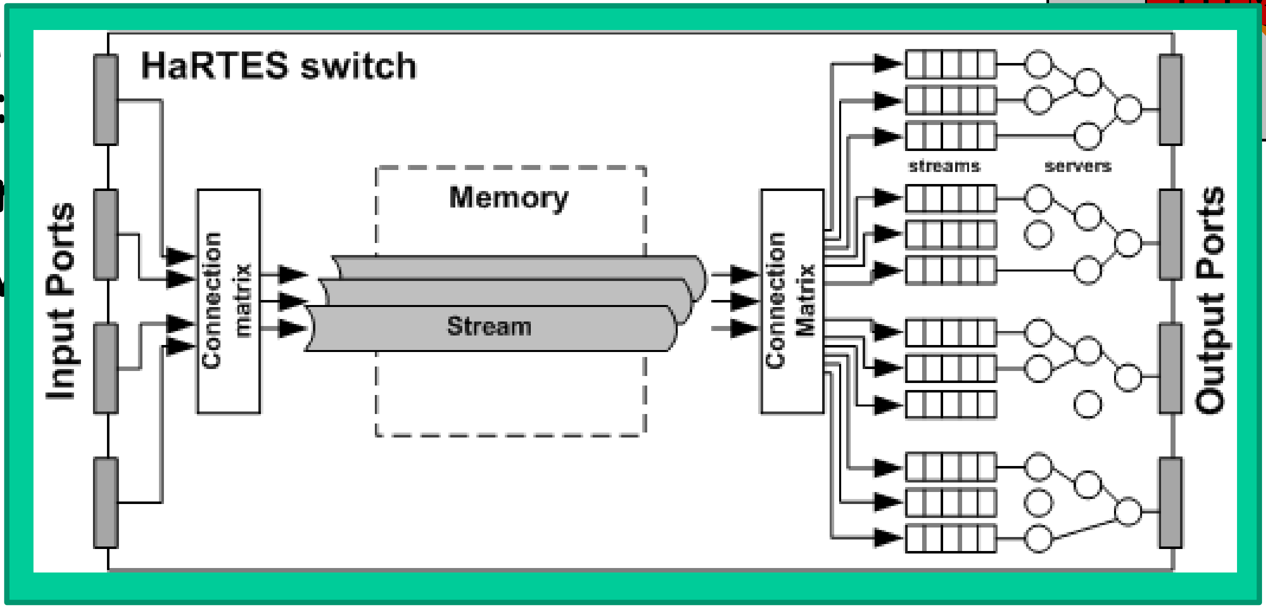
Services implementations

FTT-enabled switch

HaRTES
Hard Real-Time Ethernet Switching

- Main features:
  - Sam…
    plus:
  - **Sea…**
  - **Serv…**



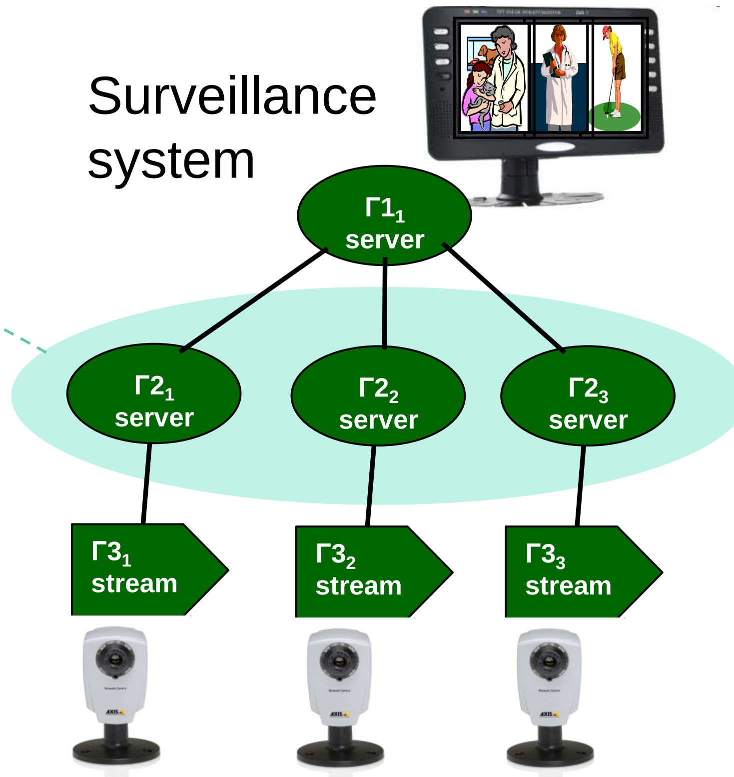


http://www.ieeta.pt/lse/hartes

Adaptive video surveillance system

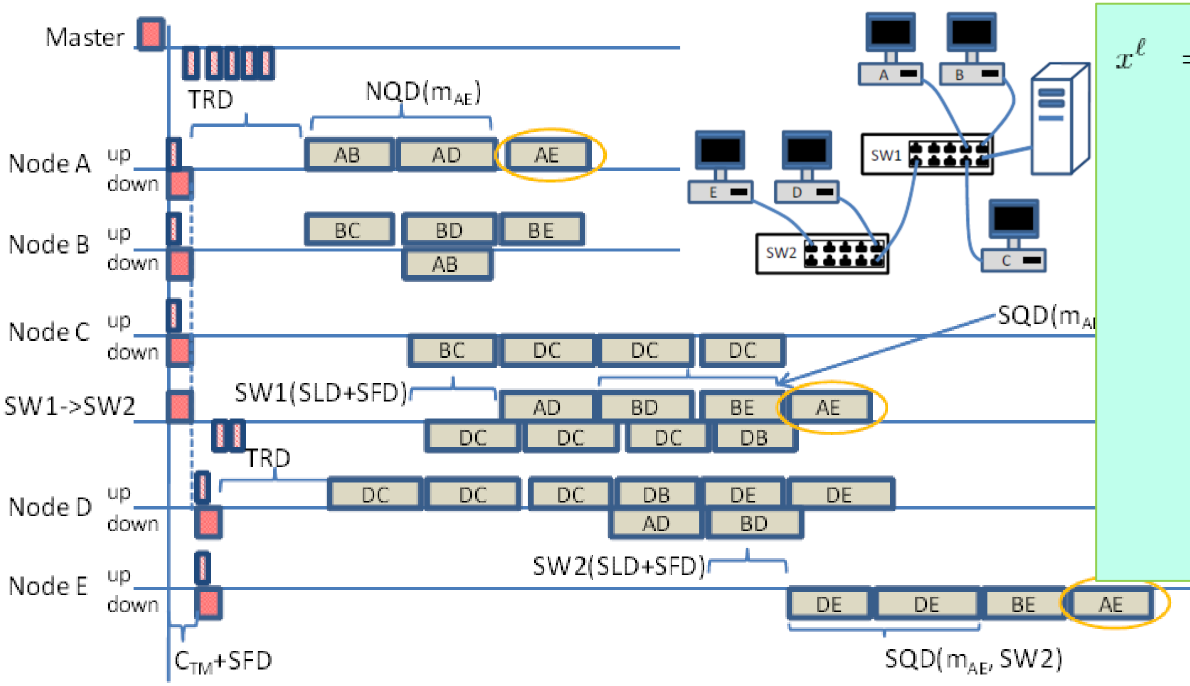
HaRTES
Hard Real-Time Ethernet Switching

- **Only 1 server** is allowed with **more BW** at a time. All others use lower BW.
- IP cameras programmed with constant frame rate
- Forces changes in frame rate due to TCP/IP sync



Scaling up these solutions

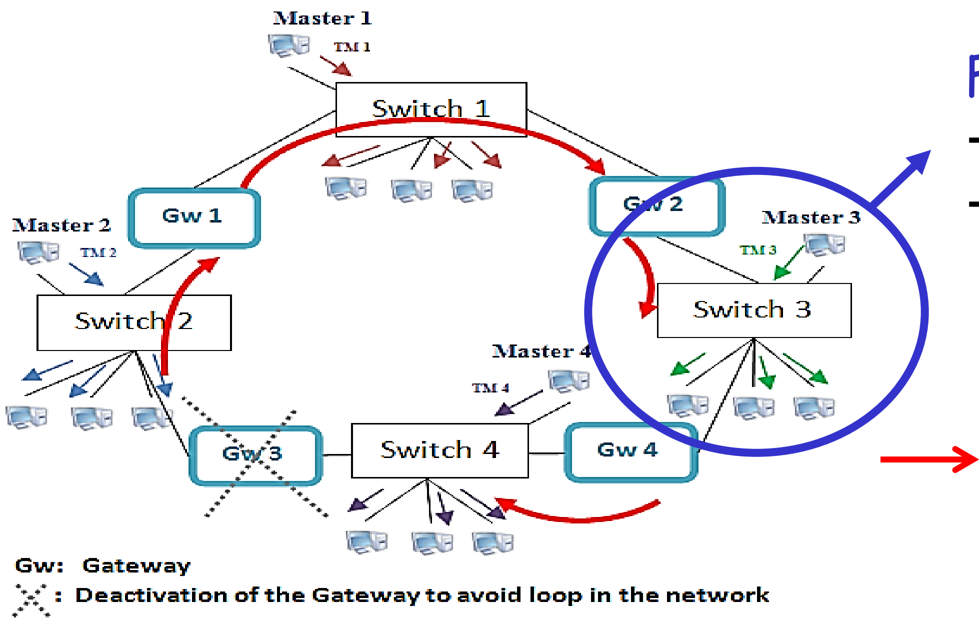
- Multiple switches per master domain (FTT-SE)
  - Not so efficient because of limited load per cycle



$$
\begin{aligned}
x^{\ell} = {} & C_i' + \sum_{SW_k \in R_i} \left(\frac{SFD(m_i,k)+SLD}{\alpha}\right) + \\
& \sum_{m_j \in S\_interf(m_i)} \left\lceil \frac{x^{\ell-1}}{P_j} \right\rceil (C_j') + \\
& \sum_{m_t \in D\_interf(m_i)} \left\lceil \frac{x^{\ell-1}}{P_t} \right\rceil (C_t' + \\
& \sum_{SW_k \in R_t} \left(\frac{SFD(m_t,k)+SLD}{\alpha}\right)) + \\
& \sum_{m_q \in I\_interf(m_i)} \left\lceil \frac{x^{\ell-1}}{P_q} \right\rceil (C_q'),
\end{aligned}
$$

Scaling up these solutions

- Single switch per master domain (FTT-SE+HaRTES)
  - Interconnection with **bridges**



FTT-SE or HaRTES
→Manage channels locally
→Provide channels globally

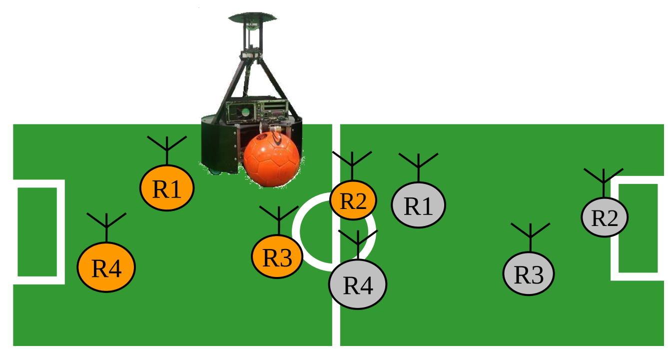
Resource reservation protocol (global channels)

On wireless networks

Target

- ✓ Support **cooperating objects**, particularly teams of **autonomous robots**

Mobile Cyber-Physical Systems



On wireless networks

Some wireless specifics

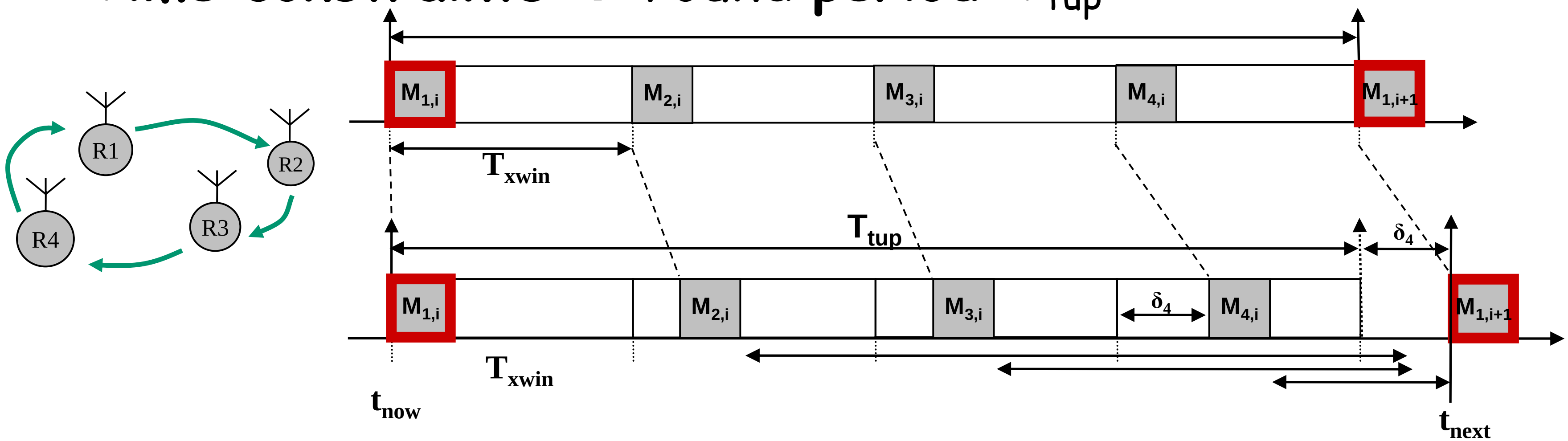
- ✓ Open medium, uncontrolled environment / load, non-stationary interference...
  - ✓ **Real-time** properties have **low coverage**
- ✓ Fading
  - ✓ **Connectivity** among the team **not guaranteed**

Our claim

- ✓ **Robots transmission** pattern is typically **periodic**
- ✓ **Automatically synchronizing** transmissions reduces chances of collision within the team
  - ✓ Improved performance mainly in **packets lost**

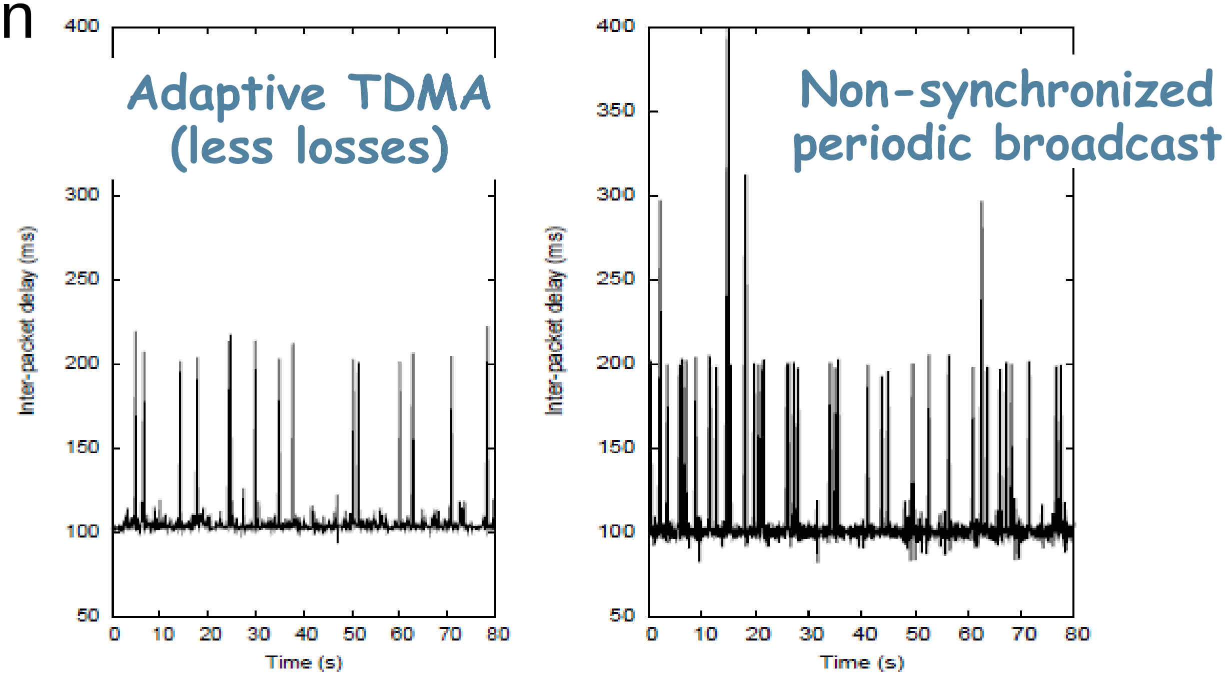
Adaptive TDMA

- **TDMA (+CSMA/CA)** with synchronization on **receptions**
  - no need for clock sync
- **Phase** of round **shifted** to match external interference
  - Maximizes separation between transmissions in the team
- **Time constraints** → round **period** T_{tup}



Adaptive TDMA

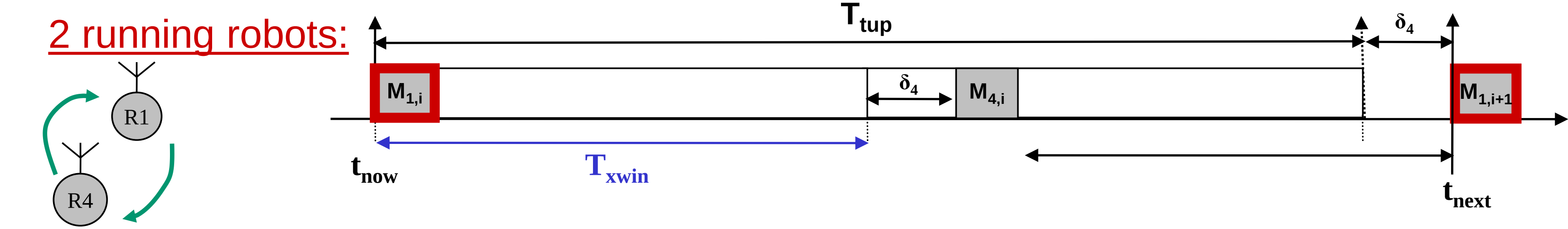
- ✓ Positive impact verified in practice
- ✓ Strongly perceived at the application level under intense communication

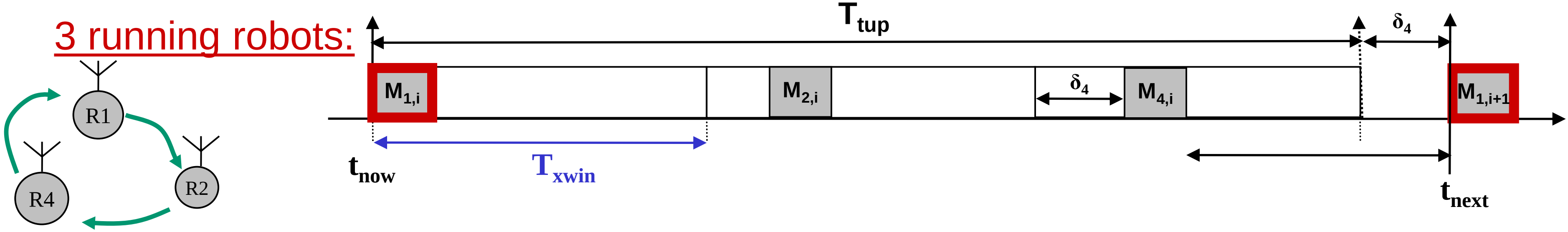


a) with sync b) without sync

Reconfigurable & Adaptive TDMA

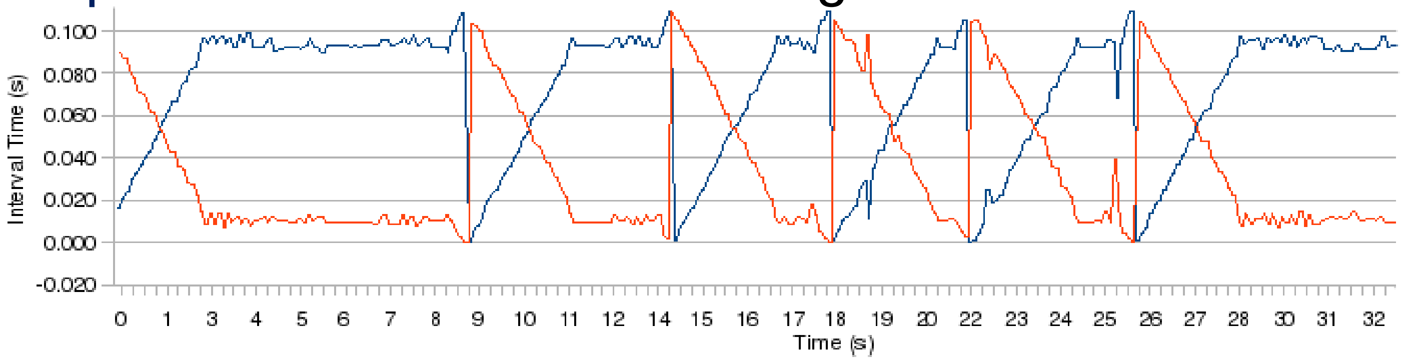
- Robots **join** and **leave** dynamically
  - crash, maintenance, movements...
- Fully distributed – **virtually configuration-free**





Reconfigurable & Adaptive TDMA

Adaptive TDMA: 2 robots running in a team of 10



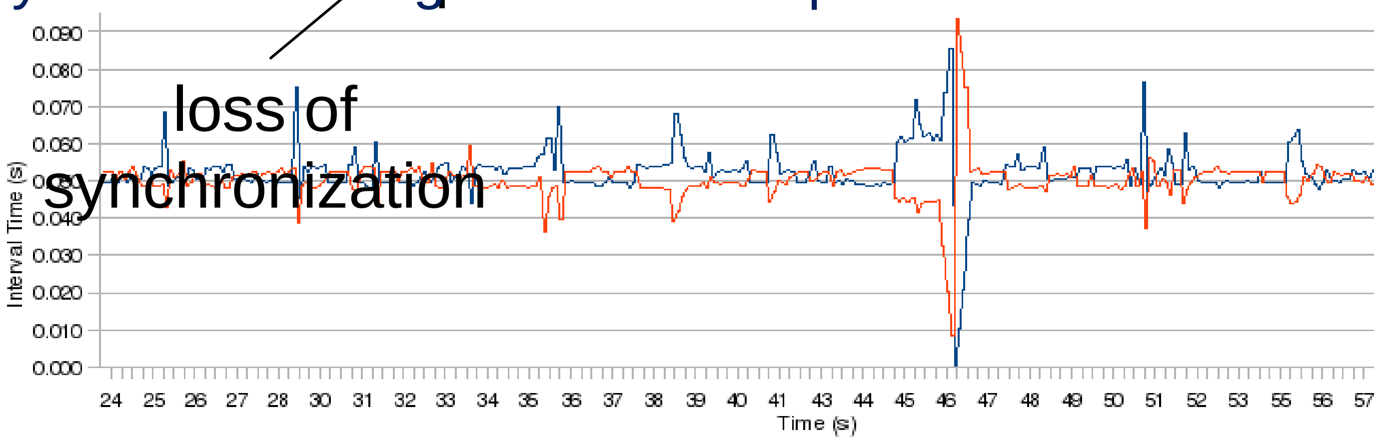
in sync: **50%**

lost packets: **1,2%**

phase rotation

Dynamic Reconfiguration of Adaptive TDMA: 2 robots running

loss of synchronization



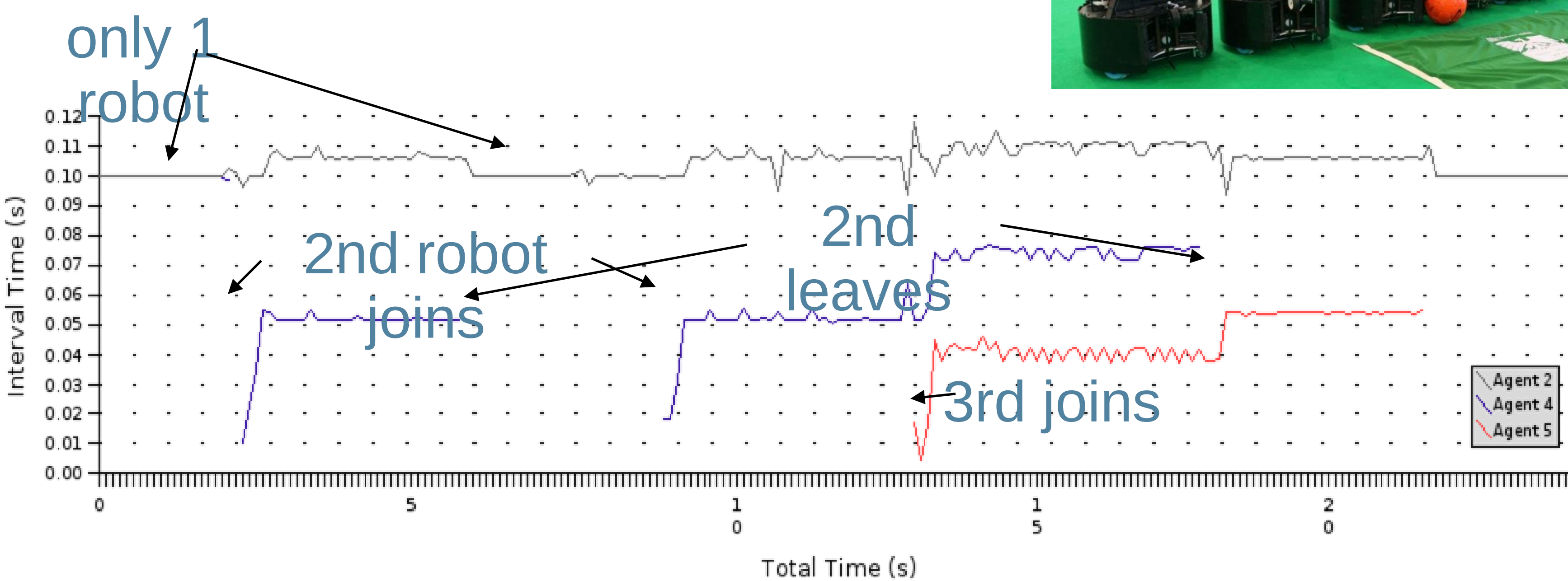
in sync: **97%**

lost packets: **0,3%**

Reconfigurable & Adaptive TDMA

- ✓ Implemented on **WiFi – infrastructured**
  - ✓ CAMBADA project, RoboCup MSL

CAMBADA



Wrapping up

- Cyber-Physical Systems are **real-time systems**!
- Generally involving **distribution / collaboration**
  - Possibly over large networks
- **Good network performance**
  (efficient, timely, reliable, secure)
  is key to emerging CPS

Wrapping up

Possible solution:

- Use **resource reservation** to partition networks in a composable way,

    particularly using

hierarchical server-based traffic scheduling

but also considering

dynamic reconfiguration and **adaptation**

Wrapping up

Two cases

1. **Good control** over ... urces
   - **Negotiate & en**... reservations **on-line**
     - Admission ... nager…
   - Good use ... **TT-SE** and the **HaRTES switch**

✓ **Law enforcement**
✓ (hard real-time)

Wrapping up

Two cases

2. **Poor control** over net ... es **(Mobile CPS)**
   - **Adaptive soft** re ... s
     - Adaptive to ...
   - Good use co ... **configurable & Adaptive TDMA**

✓**Voluntary cooperation**
✓(best effort)

Some pending issues

Law enforcement

- How **strong/robust** is the enforcing of proper resource usage?
- Control over resources and flexibility management imply **extra resource needs** (BW, CPU, energy …) !
  - Also imply extra complexity! With potential for lower reliability!
- **Global resource reservation protocol** …

Some pending issues

Voluntary cooperation

- Some kind of **guarantees** (probabilistic) would be welcome!
  - Characterization of **operational environments**
  - **Varying** communication **links**

Some pending issues

Flexibility management

- How to **distribute spare BW** among a set of users?
  - Elastic models
  - (m,k)-firm model
  - Greedy models ...
  - Act on C, on T, on both...

Some pending issues

Flexibility management (cont)

- **When** and **how** to adapt / reconfigure?
- Flexible mode changes...
- How to relate **resource usage** and **QoS**? Or even better, **QoE**?

Questions?